**530-TP-002-001**

# Communications Subsystem DFS/NFS Comparison for the ECS Project

**Technical Paper**

Technical Paper—Not intended for formal review or government approval.

**August 1995**


**RESPONSIBLE ENGINEER**

H. Naveen /s/ 8/11/95

Naveen Hota, Communications Engineer Date
EOSDIS Core System Project

**SUBMITTED BY**

Parag N. Ambardekar /s/ 8/11/95

Parag Ambardekar, Release A Manager Date
EOSDIS Core System Project



Hughes Information Technology Corporation
Landover, Maryland

This page intentionally left blank.

# Abstract

The Communications Subsystem (CSS) provides the overall communications infrastructure, and the communications services to support other subsystems in the Science and Communications Development Office (SCDO) and the Flight Operations Segment (FOS). This document describes a comparison of DFS/NFS.

***Keywords:*** CSMS, CSS, Communications, DCE, OODCE, Release A, DFS, NFS.

This page intentionally left blank.

# Contents

## Abstract

## 1. Introduction



## 2. Objectives



## 3. Requirements and Assumptions



## 4. Description of Alternatives



## 5. Evaluation



## 6. Conclusions / Recommendations



## Tables



## Abbreviations and Acronyms

This page intentionally left blank.

# 1. Introduction

## 1.1 Background

With the transition from mainframe computers to networked machines comes a desire to be able to share files between the networked machines. Early methods of sharing files involved sending entire files between machines. NFS, developed by Sun, provided an alternative approach. Rather than transferring a file from one local disk to another, NFS allowed a user to transparently access files on a remote machine. Thus, when a user accessed a part of a file, his workstation would request that part of the file from the remote machine. This allowed a change to occur in computing. Diskless workstations were invented, they are able to use another machine's disk as though it were its own, however since disk prices have dramatically dropped this is not much of an issue any more. More importantly though, it allowed users to share data even when they were operating on different machines.

However, there were limitations to what the technology could do. Currently shared disk space needs have outgrown the original NFS design. Unfortunately, the current NFS designs still maintain backwards compatibility with the original design leaving several critical flaws in the technology. Many of these are addressed by AFS (which once stood for Andrew File System, but now is no longer an acronym for anything) and DFS (which stands for Distributed File System.)

The players:

NFS was developed by Sun, and is currently the most widely used UNIX networked file system. While it has had some improvements over the past decade, the technology is still basically the same as when it was first developed. While it allows transparent sharing of files, it provides only host based security effectively - thus the only real restrictions you can place on who can read or modify files is based on what machine a user is on. While it currently provides some caching of files for faster access, it can not distinguish between part of a file being changed or the entire file being changed. For example if a user reads bytes 1-100 of a file, and her coworker modifies bytes 1024-2048, if she re-reads bytes 1-100 those bytes will have to be sent across the network again. Also, each time a user accesses a file the system either checks with the server to see if the file has changed (causing network delays) or risks the possibility that the file has changed since it was last checked. Usually a time period of 10-30 seconds is maintained, inside of which it is assumed that no files have changed. Another limitation of NFS is that each client workstation must maintain its own 'image' of how networked files are accessed, and while it is possible administratively to keep each client machine in sync with the others to provide the same view of the file system, it is not automatic. Also, if a file server is moved, each client must be re-configured in order to be able to find the new server.

AFS was developed by jointly by IBM and Carnegie Mellon University. It later became its own company called Transarc. Transarc was subsequently entirely bought by IBM and is now a fully owned subsidiary. AFS provided real security based on Kerberos authentication. It allows ACLs (Access Control Lists) to be placed on each directory specifying which users are given what types of access to the files in that directory. It also allows for AFS users in other Kerberos cells (very similar to DCE cells) to authentically access files in other cells. AFS heavily relies on caching of files for performance. When a client receives a part of a file, it saves it either in disk or memory, so that subsequent accesses to that file can be handled locally instead of requiring network traffic. It also establishes a guarantee with the file server that if the file changes, the client will be notified. This prevents the client from having to constantly poll the file server to see if there have been changes. However, when there is a change to the file, AFS must re-fetch any needed parts since it also doesn't understand about changes to one part of a file not effecting other parts of the file. In stark contrast to NFS, AFS maintains a global image of the filespace. This is maintained on the file server machines, and is automatically shared by all AFS clients. Thus, when re-configuring, moving, or adding file servers, no changes need to be made to any of the client machines, and all client machines will immediately be able to access the servers. One feature that AFS includes which also can dramatically improve performance is the idea of replication. Multiple file servers can store the same sets of data (if the data is read only), and clients will randomly select from the available servers before requesting files. This provides load balancing, improved performance and fault tolerance all at once. So, if one of the replicated file servers is down, the user's computer will automatically contact one of the other servers that is still up and continue work without interruption or failure.

DFS was designed by the OSF with contributions from Transarc. DFS is very similar to AFS, and here I will only list the differences between DFS and AFS. DFS uses DCE's security (which also happens to be based on Kerberos.) Unlike AFS, DFS provides ACLs for each file, rather than for each directory. However, currently it has the limitation that only users inside of the file server's DCE cell can have authenticated access to the files. DFS provides partial caching as well, but it is able to determine when modifications to other parts of a file do not affect currently used parts of the file and can prevent un-needed network data transmission. Currently there are other implementations of DFS (since it is an OSF standard) but Transarc/IBM are the only ones-at the time of this writing-to support ACLs and replication. However, other vendors are planning to support it in their servers in the future.

# 2. Objectives

## 2.1 Objectives

A remote file system allows remote users to access files on a disk as though the files were locally available. This provides a transparent method to transfer files when needed between machines. Also, when compared to FTP, it allows partial files to be transferred rather than requiring the entire file to be transferred.

This analysis studies features and evaluates performance of two remote file systems: DFS and NFS (V2).

This page intentionally left blank.

# 3. Requirements and Assumptions

## 3.1 Requirements

There is currently a requirement to use DFS as a pull area for Release A.

## 3.2 Assumptions

DFS and AFS require proprietary software on the clients to be able to access files, while NFS clients are standard on just about every know UNIX workstation.

This page intentionally left blank.

# 4. Description of Alternatives

## 4.1 Description

NFS (V2) servers across platforms and vendors provide the same functionality. DFS servers however offer different functions based on the vendor of the software. This evaluation will focus on Transarc's release of the DFS server for the Sparc family. Thus, for this study, the two setups looked at are:

1. An NFS based file server.
2. A DFS based file server.

This trade study initially evaluates the two alternatives as the basic underlying configuration and then evaluates the benefits gained by Transarc's DFS servers over other vendor's DFS servers, with mention of AFS. AFS is only being mentioned in passing because of the existing DCE cell architecture, it makes much more sense going with DFS over AFS.

## 4.2 NFS Based File Server

In this configuration there would be a single NFS server which could provide files to clients. Clients would have to explicitly mount the file server before being able to access the files. NFS clients are available for nearly all UNIX workstations and can optionally provide some caching of files on the client side. Most vendors have NFS servers available for their systems.

## 4.3 DFS Based File Server

In this configuration, there would be one or more DFS servers which could provide files to clients. Clients would need to be running on a machine with the DCE client software installed. File caching is automatically done. If desired users could be required to authenticate themselves before gaining access to files.

This page intentionally left blank.

# 5. Evaluation

## 5.1 Description

In evaluating the alternative solutions for the cell configuration, the following criteria were applied.

Cost

Performance

Security

Scalability

The weights in the table below are assigned as follows:

1) relatively minor

2) significant

3) critical

The score given to a server is assigned as follows:

0) Extremely poor

1) Poor

2) Average

3) Good

***Table 5-1. Cell Configuration Evaluation Matrix***

| | Weight | NFS | DFS |
|---|---|---|---|
| **Cost** | | | |
| **Software** | 2 | 2 | 1 |
| **Ease of administration** | 1 | 1 | 3 |
| **Performance** | | | |
| **Response time** | 3 | 3 | 2 |
| **Reliability of connections** | 2 | 1 | 2 |
| **Scalability** | | | |
| **Additional replicated severs** | 1 | 0 | 3 |
| **Security** | | | |
| **Security** | 3 | 1 | 3 |
| **Authentication across cells** | 3 | 1 | 2 |
| **Total** | | **22** | **33** |

In this evaluation, the following criteria were applied:

Cost

Usually NFS servers are shipped with the Vendor OS and as such, are free. DFS servers cost approximately $2,000 per server. NFS clients are usually free, and DFS clients are distributed as part of the DCE client. Thus, many platform support it (including, among others, DEC-OSF/1, HP-UX, AIX, Solaris, Cray, and this list is expanding.)

Operational cost

DFS is substantially easier to administer than NFS. Also, DFS allows delegation of certain aspects of the file system to some people while still maintaining control over the rest of the system. With NFS there is very little differentiation between different types of administration.

Performance

NFS performance is roughly twice the speed of DFS performance on a local area network under small file system load. (4 clients to 1 server) It is expected, although untested, that as the load increases DFS performance will improve and surpass the performance of NFS. DFS also allows replication of data, so that two or more servers may contain the same data set. This would allow the clients to spread the load between servers.

Scalability

Addition or renaming of NFS servers may not be done transparently. Clients must know the name of each machine that is a file server before being able to request any files from that machine. With DFS, addition of machines (and replicas of data) may be done transparently, and without any downtime for the clients.

Security

NFS, while in theory, allows group access controls to files, in practices is fundamentally insecure because it is based on an insecure protocol. DFS provides Kerberos authentication integrated with DCE. Access per file and directory may be restricted based on user, and may be administered separately from the basic DFS administration if desired.

## 5.2 DFS Variations

There are alternative venders for DFS servers for different operating systems. Currently most of them do not support ACLs or replication. Most of the venders have plans to eventually support these features, but not for a few years, which is why this study concentrated on Transarc's implementation.

Transarc also sells AFS servers and clients. AFS is similar to DFS, and in many ways was the parent of DFS. The main differences are that AFS uses vanilla Kerberos for security, rather than DCE's Kerberos, and that it only provides ACLs per directory rather than per file.

# 6. Conclusions / Recommendations

## 6.1 Conclusions and Recommendations

Based on the above evaluation, DFS is recommended over NFS for any networked file system uses. However, it is not clear if DFS is currently mature enough to trust for critical applications. It will probably be quite stable 1-2 years from now.

This page intentionally left blank.

# Abbreviations and Acronyms

| | |
|---|---|
| ACL | Access Control List |
| AFS | Andrews File System |
| AI&T | Algorithm Integration and Test |
| AIT | Algorithm Integration Team |
| ANSI | American National Standards Institute |
| API | Application program (or programming) interface |
| ASCII | American Standard Code for Information Exchange |
| ATM | Asynchronous Transfer Mode |
| ARP | Address Resolution Protocol |
| ASTER | Advanced Spaceborne Thermal Emission and Reflection Radiometer |
| BB | Bulletin Board |
| BBS | Bulletin Board Service |
| BIND | Berkeley Internet Name Domain |
| BGP | Border Gateway Protocol |
| BOA | Basic Object Adapter |
| CAC | Command and Activity Controller |
| CCB | Change Control Board (Hughes Convention) |
| CCB | Configuration Control Board (NASA Convention) |
| CCR | Configuration Change Request |
| CDS | Cell Directory Service |
| CDR | Critical Design Review |
| CDRL | Contract data requirements list |
| CERES | Clouds and Earth's Radiant Energy System |
| CIDR | Classless Interdomain Routing |
| CM | Configuration management |
| CMAS | Configuration Management Application Service |
| CMIP | Common Management Information Protocol |
| CNE | Campus Network Environment |

| | |
|---|---|
| CORBA | Common object request broker architecture |
| COTS | Commercial off-the-shelf (hardware or software) |
| CPU | Central processing unit |
| CSMS | Communications and System Management Subsystem |
| CSS | Communication Subsystem |
| DAAC | Distributed Active Archive Center |
| DADS | Data Archive and Distribution System |
| DB | Database |
| DBMS | Database management system |
| DCE | Distributed computing environment (OSF) |
| DEC | Digital Equipment Corporation |
| DECOM | FOS Decommutation Process |
| DFS | Distributed File System |
| DID | Data item description |
| DME | Distributed Management Environment |
| DNS | Directory Name Service |
| DOF | Distributed Object Framework |
| DPR | December Progress Review |
| DS | Data Server (FOS) |
| DTS | Distributed Time Server (part of DCE) |
| ECS | EOSDIS Core System |
| EDOS | EOS Data and Operations Center |
| EDF | ECS Development Facility |
| E-Mail | Electronic Mail |
| EMC | Enterprise Monitoring and Coordination |
| EOC | EOS Operations Center (ECS) |
| EOS | Earth Observing System |
| EOSDIS | Earth Observing System Data and Information System |
| EP | Evaluation Prototype |
| ESN | EOSDIS Science Network |
| EPV | Endpoint Vector |

| | |
|---|---|
| FDDI | Fiber distributed data interface |
| FDF | Flight Dynamics Facility |
| FOS | Flight Operations Segment |
| Ftp | File Transfer Protocol |
| GB | Gigabyte ($10^9$) |
| GCDIS | GDS Global Directory Service |
| GDS | Global Directory Service |
| GSFC | Goddard Space Flight Center |
| GUI | Graphic user interface |
| HAIS | Hughes Applied Information Systems (ECS) |
| HiPPI | High Performance Parallel Interface |
| HP | Hewlett Packard |
| Http | Hyper Text Transfer Protocol |
| I/F | Interface |
| I&T | Integration & Test |
| IBM | International Business Machines, Inc. |
| ICD | Interface control document |
| ICMP | Internet Control Messaging Protocol |
| IDL | Interface Definition Language |
| IEEE | Institute of Electrical and Electronics Engineers |
| IETF | Internet Engineering Task Force |
| IP | Internet Protocol |
| IR-1 | Interim Release 1 |
| ISO | International Standards Organization |
| ISO+ | IsoCELL (Isolation Cell) |
| ISS | Internetworking Subsystem of CSMS |
| IST | Instrument Support Toolkit |
| IST | Instrument Support Terminal |
| Kerberos | Security protocol developed by MIT; base for DCE security |
| Kftp | Kerberized file transfer protocol |
| KLOC | Kilolines ($10^3$) of code |

| | |
|---|---|
| Ktelnet | Kerberized telnet |
| LAN | Local area network |
| LaRC | Langley Research Center |
| LLC | Logical Link Control |
| LOC | Lines of code |
| LSM | Local System Management |
| M&O | Maintenance and operations |
| MBONE | Multicast Backbone |
| MIB | Management Information Base |
| MIME | Multimedia Internet Mail |
| MLM | Mid-Level Manager |
| MOPITT | Measurement of pollution in the troposhere |
| MOSPF | Multicast Open Shortest Path First |
| MR-AFS | Multi-Resident Andrew File System |
| MSFC | Marshall Space Flight Center |
| MSS | Systems Management Subsystem |
| MUI | Management User Interface |
| NCR | Non-conformance Report |
| NFS | Network file system |
| NIC | Network Interface Card |
| NNTP | Network New Transfer Protocol |
| NOAA | National Oceanic and Atmospheric Administration |
| NOLAN | Nascom Operational Local Area Network |
| NSI | NASA Science Internet |
| NTP | Network Time Protocol |
| OA | Off-Line Analysis Process |
| OLAP | On-Line Analytical Processing |
| OLTP | On-Line Transaction Processing |
| OMG | Object Management Group |
| OMT | Object Modelling Technique |
| OO | Object-oriented |

| | |
|---|---|
| OODCE | Object-oriented DCE |
| OODBMS | Object-oriented database management system |
| ORB | Object Request Broker |
| OS | Object Services (CSS building blocks) |
| OSF | Open Software Foundation |
| OSI | Open System Interconnect |
| OSI-RM | OSI Reference Model |
| OSPF | Open Shortest Path First |
| PAC | Privilege Attribute Certificate |
| PDR | Preliminary Design Review PDR-A |
| PDU | Protocol Data Unit |
| PPP | Point-to-Point Protocol |
| POSIX | Portable Operating System Interface for Computer Environments |
| PSC | Pittsburgh Supercomputing Center |
| PTGT | Privilege Ticket Granting Ticket |
| RDBMS | Relational database management system |
| RFA | Remote File Access |
| RFC | Request for comments |
| RIP | Routing Information Protocol |
| RMA | Reliability, Maintainability, Availability |
| RMON | Remote Monitoring |
| RMP | Reliable Multicast Protocol |
| RPC | Remote procedure call |
| RTS | Real-Time Server (FOS) |
| SCF | Science Computing Facility |
| SDPF | Sensor Data Processing Facility |
| SDR | Software/System Design Review |
| SDR | Sensor data record |
| SGI | Silicon graphics |
| SLOC | Source lines of code |
| SMC | System Monitoring and Coordination |

| | |
|---|---|
| SMDS | Switched Multi-megabit Data Service |
| SMTP | Simple Mail Transfer Protocol |
| SNMP | Simple Network |
| SQL | Simple Query Language |
| TBD | To be determined |
| TCP/IP | Transmission Control Protocol/Internet Protocol |
| TGT | Ticket Granting Ticket |
| TMN | Telecommunications Management Network |
| TRMM | Tropical Rainfall Measurement Mission |
| TSDIS | TRMM Science Data Information System |
| UDP | User Datagram Protocol |
| UIOAR | User Interface Off-Line Analysis Request Window |
| URL | Universal Resource Locator |
| US | User Station (FOS) |
| UUID | Universal Unique Identifier |
| UTC | Universal time code |
| V0 | Version 0 |
| VT | Virtual Terminal |
| WAN | Wide area network |
| WWW | World Wide Web |
| X | X Protocol |
| X.500 | OSI standard for directory services (207) |
| XDS | X/Open Directory Service |
| XFN | X/Open Federated Naming |
| XOM | X/Open OSI-Abstract-Data Manipulation |